July 2020

Encryption and Quantum Computing

Robin Wilton
wilton@isoc.org

First, the disclaimer:

“If you think you understand quantum mechanics, you don’t understand quantum mechanics.”

Prof. Richard Feynman (Nobel physicist, pioneer in sub-atomic particles and quantum computing, and world-class explainer of hard stuff.)

What follows makes heavy use of metaphors…

Main goals of this presentation

- To understand enough about encryption and quantum computing to gauge whether the latter represents a fatal threat to the former.
- To understand what, if anything, we can do to mitigate any such risk.

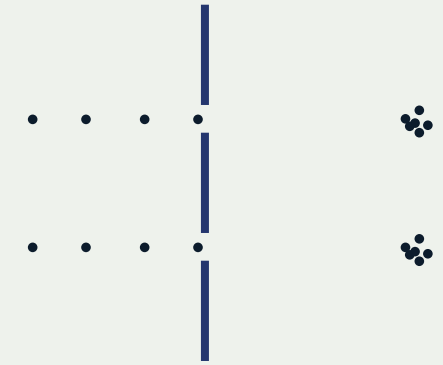
Topics

- A quantum of physics
- A bit of encryption
- A qubit of quantum cryptanalysis
- Some closing thoughts

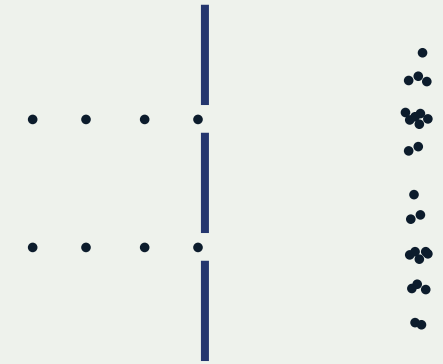
A Quantum of Physics

- Classical physics cannot explain some of the things we observe in the universe* - for instance, how light can seem to act as both a wave and a particle simultaneously.
- But quantum physics can explain these phenomena at a sub-atomic level.
- In the case of light, quantum physics says that photons “superpose” different states at the same time - a wave-like state and a particle-like state.
- (If light were a wave, you would expect to see an oscillating point of light at the point where it reaches the back-screen.)

*For more examples, I recommend Prof. Jim Al-Khalili’s amazing programme on Quantum Biology, “Let There Be Life”: (https://www.youtube.com/watch?v=q4ONRJ1kTdA)



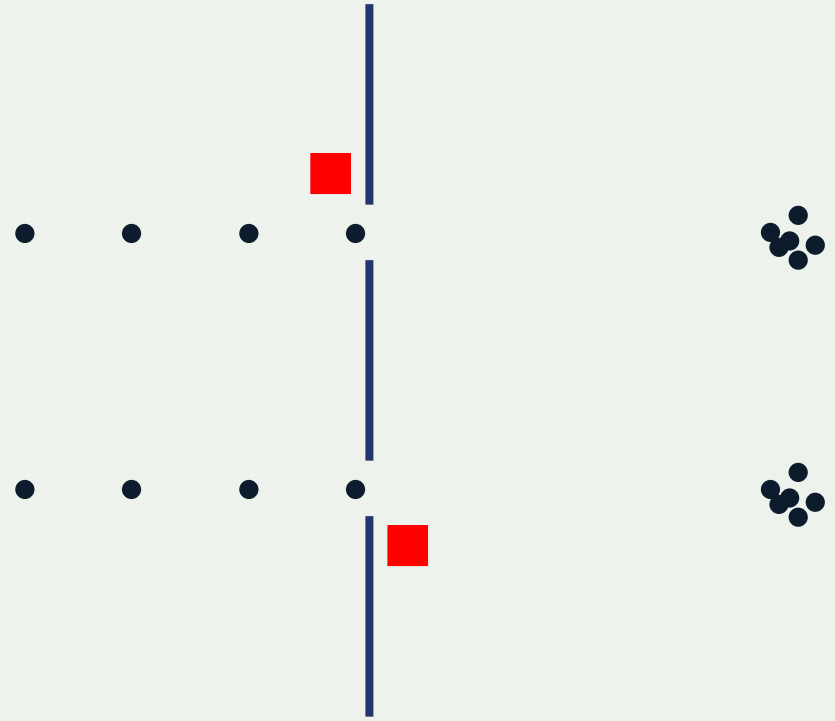
A: How a beam of photons (particles) should behave
A: How a beam of photons (particles) should behave



B: Photons exhibiting wave-like interference patterns

Superposed states

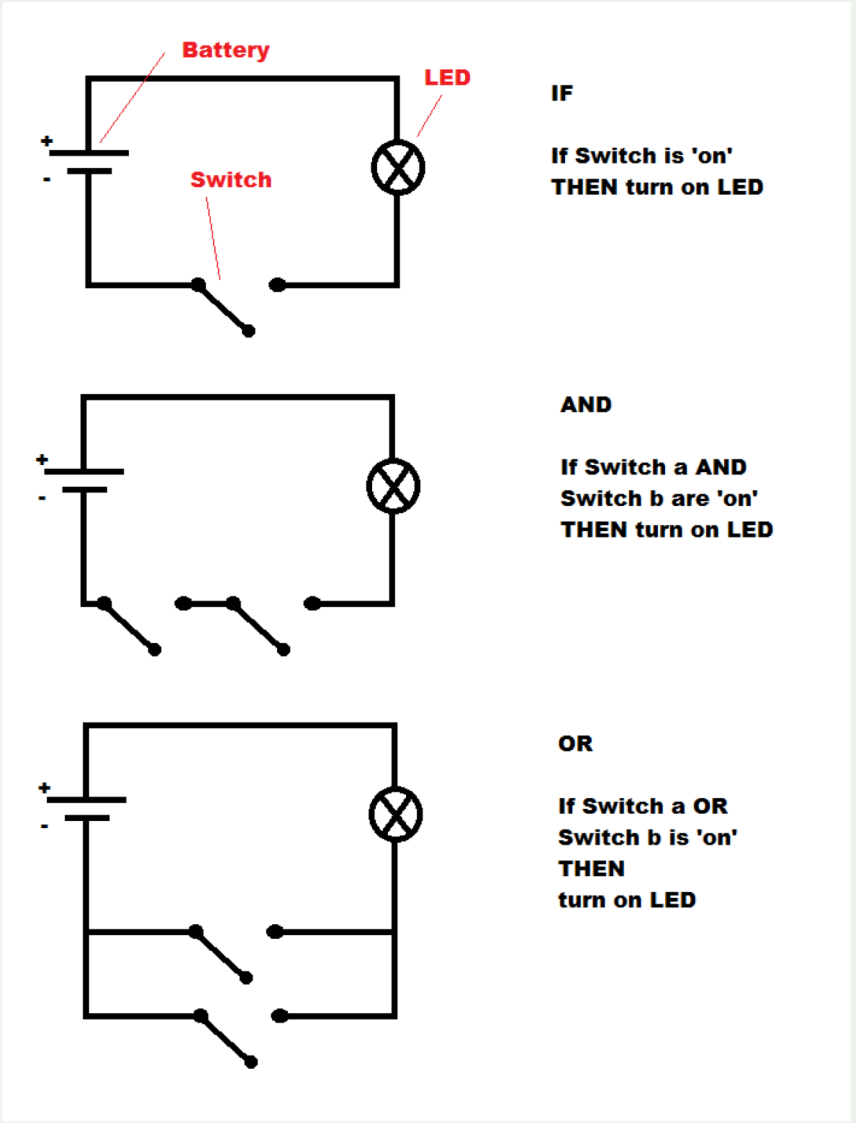
- The “dual slit” experiment is designed to show that photons can superpose a wave state and a particle state.
- It also shows that quantum physics is fundamentally probabilistic in nature:
  - we can observe an interference pattern “after the event”
  - (weirdly) before that observation, the only information accessible about which path a photon takes to reach the backscreen is probabilistic…
  - A detector placed in front of or behind the slits , to determine which slit a photon actually goes through, causes the interference pattern to disappear.



Classical computing

- Classical (digital) computing is based on ones and zeroes:
  - A bit can store a value of 1 or 0
  - Using binary values, we can do arithmetic and we can also construct switches.
  - Combining binary switches gives us logic gates: AND, OR, IF-THEN, etc..
  - Think of a light in your home with dual switching (an OR gate), or a dual-key missile launch button (an AND gate).

- Quantum computing changes that model…

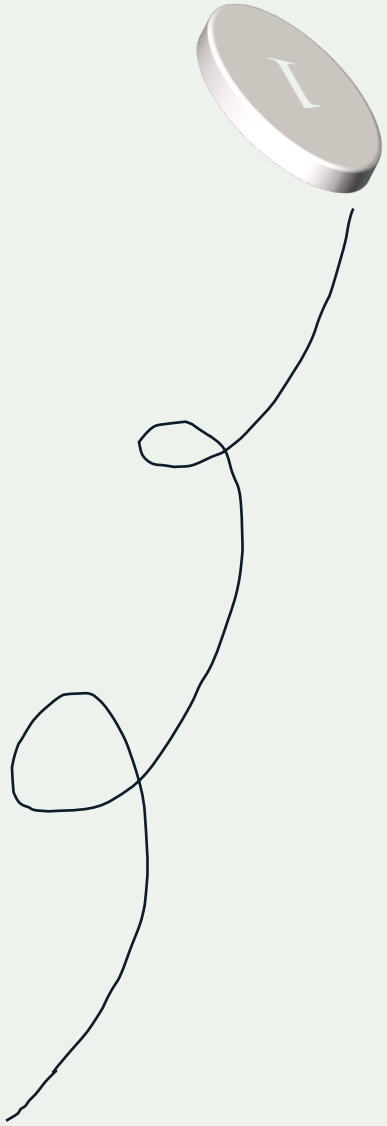


Source: http://www.thebioneer.com/wp-content/uploads/2017/04/Logic-Gates.png

Quantum computing

- Quantum computing is the applications of quantum physics to computer processing. Specifically, it applies the idea of superposition to bits, allowing them to superpose the states of 1 and 0…
- … a qubit.
- This is rather like flipping a coin; you know it must eventually land in one of three states, but until it does so, it is “superposing” all three – or, at least, superposing a set of probabilities of its end state.
- When Schroedinger applied this principle to his (hypothetical) cat-in-a-box, the thought experiment was this:
    1. The cat’s death is determined by a quantum event;
    2. The event happens;
    3. The state of the event is observed.

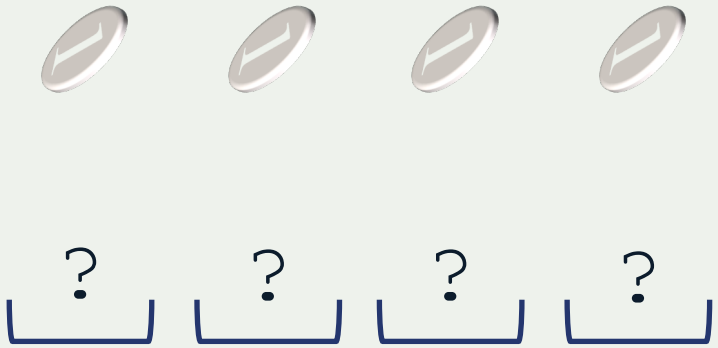
  The phrase Schroedinger used was that, until the state of the event is observed, "the living and dead cat (pardon the expression) [are] mixed or smeared out in equal parts”. Its properties (alive/dead) have many values at the same time, and all we can do is to assign them a probability.



Quantum computing

- To illustrate the potential of quantum computing, let's take a trivial example.
- Suppose you have a really basic computer, with 12 bits of storage, an add function and a compare function, and you want to do some arithmetic on two 4-bit binary numbers. Here's the sum we want to do:

```
0 0 0 1
        +
? ? ? ?
        =
1 0 0 0
```



- Using binary bits, you'd have to run through 8 possible values of the missing number, one after another, until you found the correct one.
- If we had 4 qubits to work with instead, each qubit could superpose the values of 1 and 0, and therefore all the possible values of the missing number, simultaneously.

Quantum computing

- The snag, of course, is that each qubit can only superpose those states until we observe them, at which point - like the photon suddenly acting like a particle rather than a wave – they “collapse” into either a 1 or a 0.
- Sean Ong has made a great video to illustrate the “spinning coins” metaphor https://www.youtube.com/watch?v=lypnkNm0B4A
- … up to the point where we encounter that problem: how to get the correct answer from the “collapsed” states of our observed qubits.
- tl;dr: it’s hard. Hard to understand, hard to do, hard to explain. The answer is based on sorting algorithms and probability weightings. The Wikipedia article on “Grover’s Algorithm” has more detail but quickly enters a realm where the mathematics contains more letters than numbers…
- The bottom line is that if a classical computer has to try N possible values to be sure of finding the correct one, Grover’s algorithm reduces that task to $\sqrt{N}$, which, in the encryption domain, is a significant change.
- Let’s see why…

Topics

- A quantum of physics
- A bit of encryption
- A qubit of quantum cryptanalysis
- Some closing thoughts

Symmetric encryption

- Symmetric encryption scrambles data, in combination with a secret key, such that the original data can be recovered by reversing the scrambling process with the same key.
- It works like a cash box: whatever key is used to lock it, the box can be unlocked with that key or an exact copy. (But you still have the problem of sharing the key securely...)

- A good symmetric encryption algorithm is designed so that, in the absence of the secret key, there is no more efficient way to recover the original cleartext than by trying every possible key on the ciphertext until you hit the right one – an exhaustive or brute force attack.
- If the number of possible keys is big enough, the chances of finding the correct one *at random* are negligible, and the task of finding the correct one *systematically* is described as “computationally infeasible”.

- In “Applied Cryptography”, Bruce Schneier sets out, in terms of pure physics, what would be involved in an exhaustive attack on 256-bit symmetric keys. (2nd edition, pp. 157-8), or here, on his blog: https://www.schneier.com/blog/archives/2009/09/the_doghouse_cr.html

Symmetric encryption

- The previous slide referred to brute-force attacks. To quantify the threat from quantum computing, we need to look at keyspace and work factor.
- The keyspace is the total number of possible keys for a given key length. For binary keys, this number is 2^n, where n is the key length in bits. (So, our simple 4-bit computer earlier had a keyspace of 2^4, i.e. 2x2x2x2 = 16 possible keys.)
- Each time you add a bit to the key length, you double the keyspace.
- If you double the key length, the keyspace is squared.
- The effort required for an exhaustive search is referred to as the "work factor".
- Work factor is quantifiable in terms of cycles, time and money – but ultimately, matter and thermodynamics.

Some big numbers...	
2^{37}	Number of stars in our galaxy
2^{77}	Approx. stars in the observable universe.
2^{92}	Mass of the Earth, in grams
2^{170}	Atoms in our planet
2^{223}	Estimated atoms in our galaxy
2^{256}	Keyspace of your browser's TLS key

Asymmetric encryption and key distribution

- Symmetric algorithms suffer from the problem of secure key distribution. If you send someone a message in a locked box, how do you securely get the key to them?

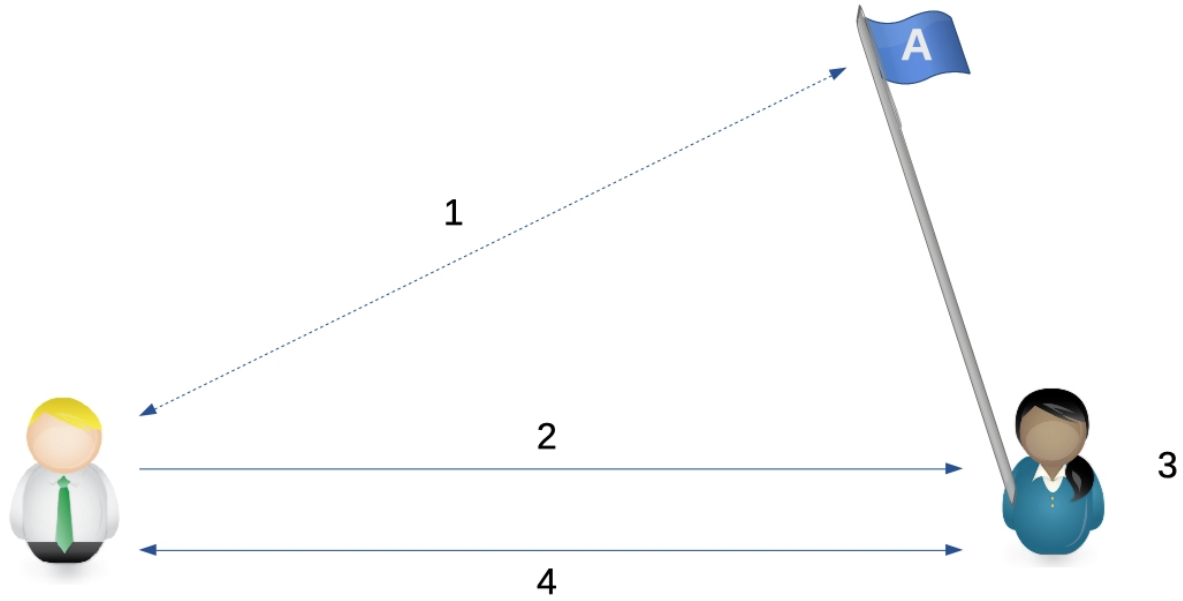
- Asymmetric or public-key encryption offers a solution to this problem.

- Each user has a pair of keys: a public one for encryption and a private one for decryption. To send Bob a message, Alice looks up Bob's public key and encrypts her message with it. Bob decrypts the message with the corresponding private key, which only he knows.
- Crucially, unlike symmetric encryption, "reversing" the encryption function, with Bob's public key as input, does not recover the plaintext.

- If Alice's message is, in fact, the secret key for a symmetric algorithm, we've achieved secure key distribution – hybrid systems like this, in the form of TLS, are the most widely-deployed encryption technology on the planet.

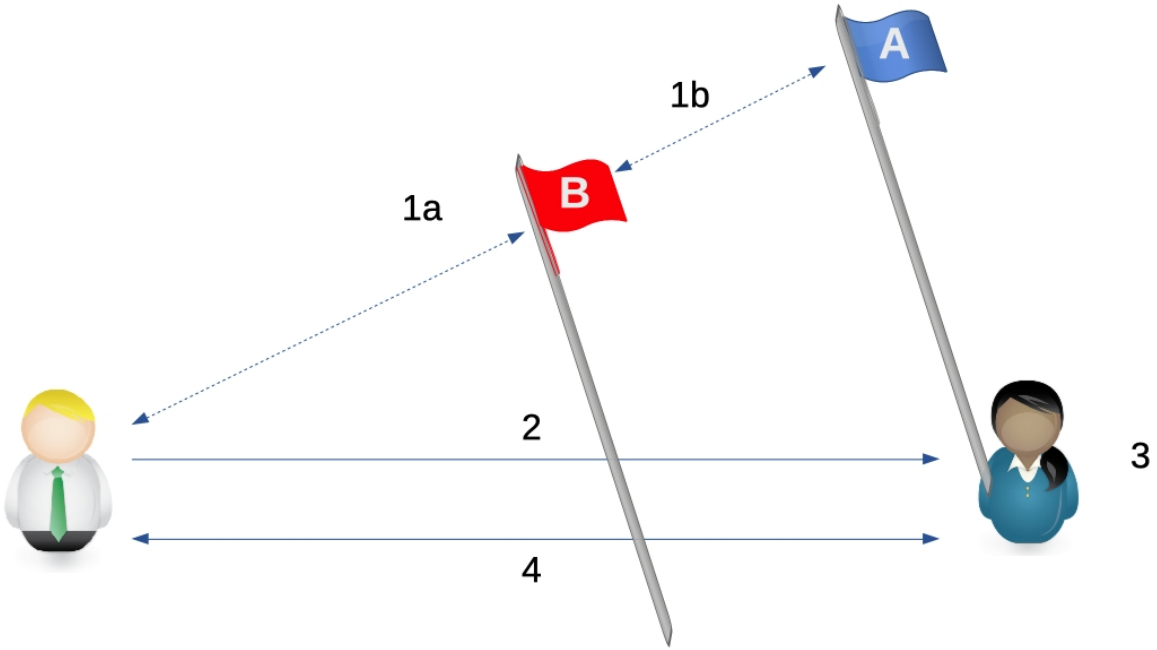
- Guaranteeing that Bob's public key really belongs to him relies on a series of digital signatures, making them an interesting target for cryptanalysis.

Public key encryption: initial key exchange.



1. Gerald looks up Leila's public key [A] and copies it to his device.
2. He generates a secret (symmetric) key [S], encrypts that under Leila's public key, and sends it to her.
3. Only Leila can decrypt the message (using her privacy key [B])
4. Gerald and Leila now both have a copy of [S].

Public key encryption: “Man In The Middle” attack.



1a: Gerald thinks he’s looking up Leila’s public key [A], but Max has managed to place his ”flag” (key) in front of Leila’s.. Max replies to Gerald with an apparently valid key.

1b Meanwhile, he sends a request to Leila, pretending it’s from Gerald.

2-3 Max now has one shared key pair with Gerald, and another with Leila.

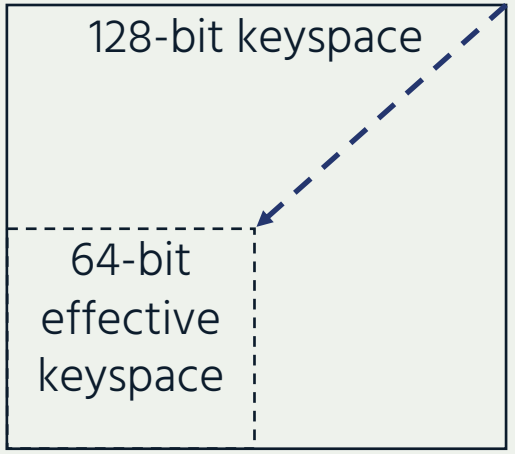
4: Gerald and Leila both think they are talking to each other, but Max is intercepting everything,, reading it, and re-encrypting it to forward.

Topics

- A quantum of physics
- A bit of encryption
- A qubit of quantum cryptanalysis
- Some closing thoughts

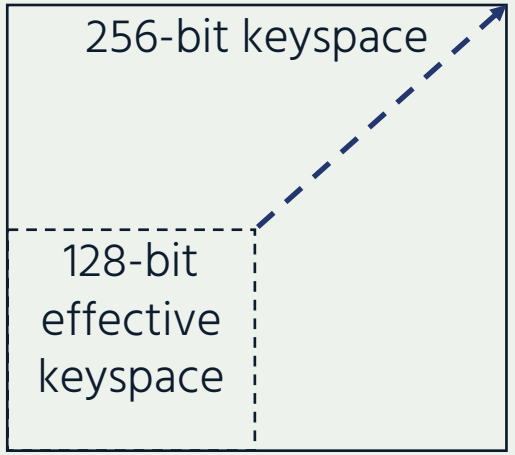
Defending symmetric encryption against quantum cryptanalysis

- A hybrid system combining quantum computing, Grover's algorithm and classical computing reduces the effective keyspace to its square root.



- The simple* answer is to double the key length (thus squaring the keyspace again).

*replacing deployed encryption systems is known to be slow, especially if you need all your communicating partners and devices to do it...

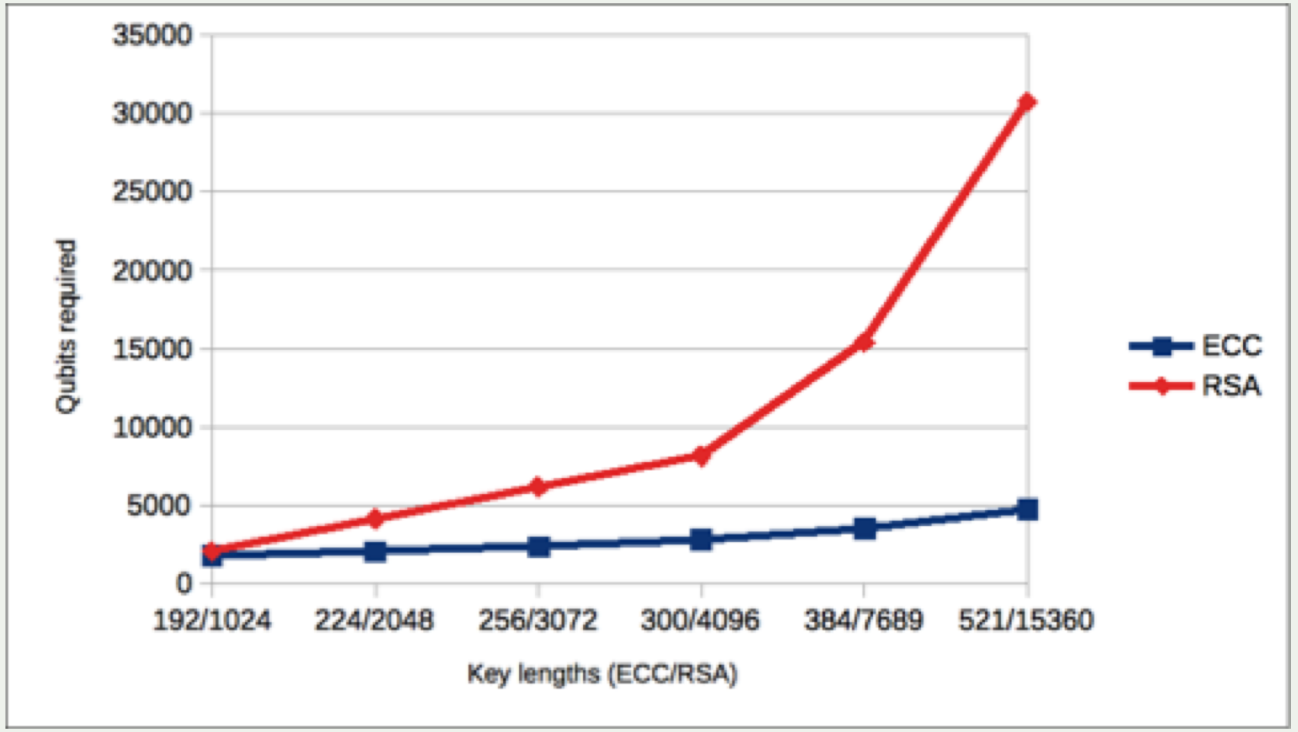


Quantum cryptanalysis of asymmetric (public key) encryption

- The most common public key algorithms in use are RSA and elliptic curve cryptography.
- These are based on the mathematical difficulty of factoring large primes (RSA) or solving discrete logarithm problems (ECC).

- Shor's algorithm reduces the difficulty of these problems to the point where they can be solved "in polynomial time" – which means they are no longer complex enough to deliver safe encryption – but it does need a lot of qubits…

- For more about "polynomial time" as a measure of difficulty, see
  https://en.wikipedia.org/wiki/Complexity_class
  https://en.wikipedia.org/wiki/P_(complexity)



Topics

- A quantum of physics
- A bit of encryption
- A qubit of quantum cryptanalysis
- Some closing thoughts

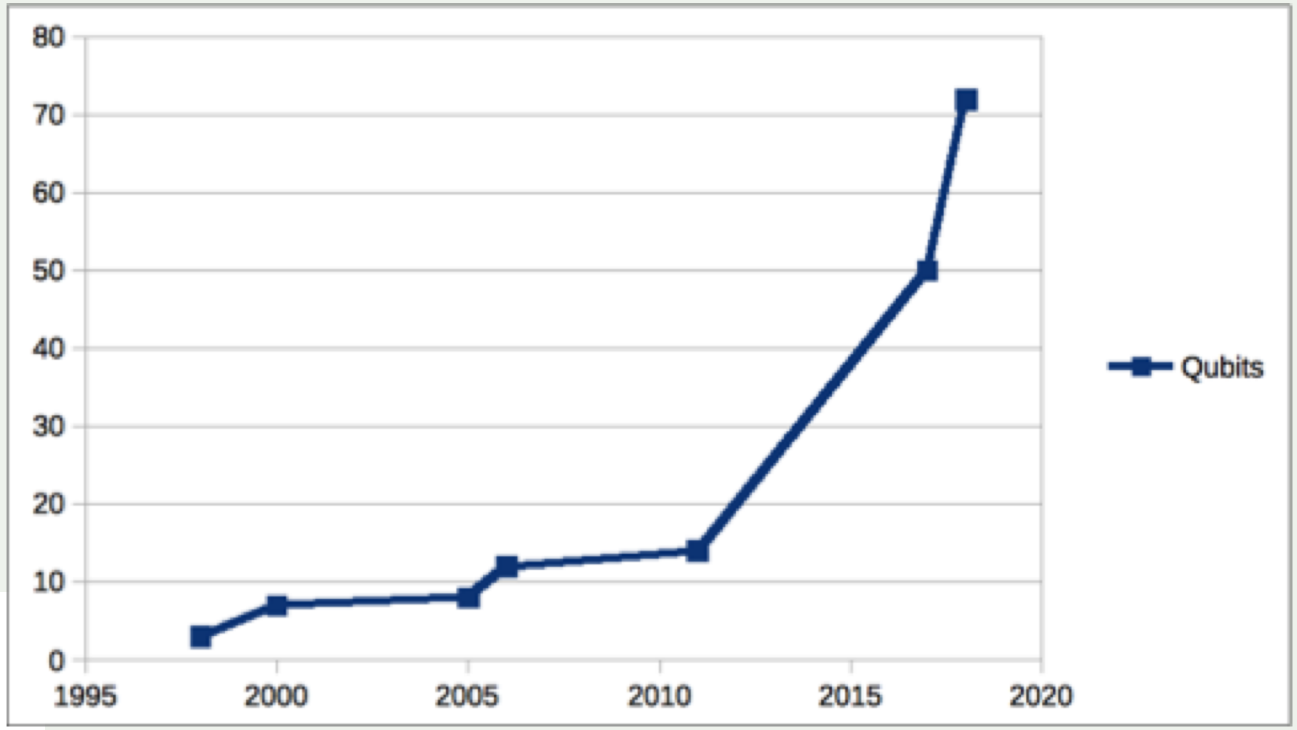
Topics

- Feasibility; numbers of working qubits
- Progress towards quantum-resistant algorithms
- Other domains of quantum computing
- Recommendations

Feasibility

- This may look scary, but it still falls short of the scale needed for a viable attack.

- Bear in mind also the ratios of key bits to qubits…
  - 1:1 for symmetric encryption, but
  - 2:1 for RSA, and
  - 9:1 for ECC



The number of qubits is only part of the problem; they tend to "decay", and are easily perturbed by external factors (including each other!).
Here's an indication of progress towards stable persistence of qubits…

2009	"hundreds of milliseconds"	i.e. 10ths of a second
2013	39 minutes	(at room temperature)
2017	90 microseconds	… but 50 qubits at once

Post-Quantum Cryptography (PQC)

- Advances in quantum computing have spurred research into mathematical problems that are hard even for quantum computers to solve.
- NIST has just announced a set of candidate algorithms for the next round of the standardization process: https://csrc.nist.gov/publications/detail/nistir/8309/final

- NIST also has more PQC-related resources here:

https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

- It seems likely that classical computing will be able to implement harder algorithms faster than quantum computing can break them – but that's no good if you're still on an obsolete algorithm.

Quantum computing is not a single technology

- Quantum computing takes many forms, not all of which are applicable to cryptanalysis:
  - Quantum metrology
    - Sensor technology
    - Navigation
    - Time sources
  - ‘General purpose’ quantum computers
  - Quantum computers for cryptanalysis

- General purpose quantum computers are unlikely to be the most efficient tools for cryptanalysis, and as we have seen, resource constraints soon become a factor.

- Greater scale and efficiency can be achieved by designing for a particular problem, but the resulting system will then be specific to one algorithm, which again increases cost.

- Quantum cryptanalysis is likely to remain a specialized domain, of interest to some specific stakeholder types.

What can/should we do?

- As individual users, probably not much.
- Governments, enterprises, infrastructure providers and other organisations should:
  - Monitor this space for inflection points
  - Track progress towards PQC
  - Factor PQC into their risk analysis: what would happen if they suddenly had to re-encrypt all their securely archived data? What would happen if a digital signature algorithm suddenly became unreliable for signing, authentication and key exchange?
  - Do what they can to maximise algorithm agility (for instance, hybrid signature systems).

- A quantum of physics
- A bit of encryption
- A qubit of quantum cryptanalysis
- Closing thoughts

Thank you –

Any questions?

July 2020

Encryption and Quantum Computing

Robin Wilton
wilton@isoc.org